

Parcial 2023

Programación para Ingeniería Eléctrica

7 de julio de 2023

La prueba tiene un total de **30** puntos. El mínimo requerido es de **8** puntos (25% del puntaje total de la prueba). El tiempo máximo para responder este parcial es de 3 horas. Respondan cada ejercicio en una hoja diferente, la cual debe contener su nombre, cédula y número de hoja.

Ejercicio 1 - Argumentos de línea de comando, pasaje de parámetros, impresión en pantalla (10 puntos)

a) Declare el encabezado usual de la función `main` de un programa llamado `matriz` que tome una lista de argumentos de la línea de comandos y devuelva un `int` como resultado.

b) Escriba el código de dicha función `main` que haga lo siguiente:

- Declarar un puntero a caracteres sin signo `M`. En este parcial `M` va a ser un puntero simple capaz de almacenar una matriz de $m \times n$ valores, tal como se utilizó en el obligatorio.
- Lea el primer argumento de la línea de comandos `y`, considerando que se trata de un entero, lo convierta y con su valor inicialice la variable `m`. Recuerde que debe incluir la biblioteca `stdlib`.
- Lea el segundo argumento de la línea de comandos `y`, considerando que se trata de un entero, lo convierta y con su valor inicialice la variable `n`.
- Llame a una función `reservaMemoriaMatriz`, pasándoles como argumentos los valores de `m` y `n` leídos de la línea de comando y `M` pasado por referencia.
- Si la función `reservaMemoriaMatriz` devuelve 0, debe imprimir en pantalla el contenido de `M`. Se debe escribir como una matriz, de tal manera que si es una de dimensión de 3×2 y su contenido fuera $i + j$ se vería así:

```
0 1
1 2
2 3
```

Ninguna fila tiene espacio luego del último término. Luego del último carácter numérico impreso debe haber solo un salto de línea.

- Si la función `reservaMemoriaMatriz` devuelve un valor distinto de cero, debe imprimir en pantalla un mensaje que diga `error` y salte a la línea.
- Salga del programa.

Ejercicio 2 - Reserva de memoria dinámica y manejo de arreglos unidimensionales para guardar una matriz bidimensional (10 puntos)

Escriba la función `reservaMemoriaMatriz` que se llama en el ejercicio anterior. Esta función debe devolver un entero y recibe como argumentos:

- un entero m .
- un entero n .
- un puntero a caracteres sin signo pasado por referencia `pM` (es decir, `pM` es un puntero doble).

Dicha función debe:

- Reservar suficiente memoria para la matriz pasada por referencia de dimensiones $m \times n$ de caracteres sin signo. Si no se pudo reservar memoria, por cualquier causa, devolver -1 .
- Modificar el contenido de la matriz de forma tal que cada posición tenga la suma de la posición de su fila y columna, es decir, $M[i, j] = i + j$, empezando a numerar las filas y columnas desde 0.
- Devolver el valor 0 indicando que la función terminó de ejecutarse sin dificultades.

Ejercicio 3 - Manejo de bits (4 puntos)

En este ejercicio solo se trabajara sobre dos variables `unsigned char x, y`. Solo se permite utilizar operaciones de bit sobre estas variables y se supone que el bit 0 es el menos significativo y está a la derecha.

a) Escriba una línea de código que guarde en la variable `y` el valor del bit 3 de la variable `x`.

```
y = (a completar...);
```

b) Escriba una línea de código que guarde en la variable `y` la variable `x` con su quinto bit modificado a 1.

```
y = (a completar...);
```

Ejercicio 4 - Compilación (4 puntos)

Suponga que tenemos un archivo `programa.c` que a su vez usa funciones definidas en otro archivo `foo.c` y una serie de estructuras.

a) Explique por qué es conveniente tener un archivo `foo.h` y qué debe incluir el mismo. Además explique cómo se hace el `include` desde `programa.c`.

b) Suponga ahora que en cada uno de los archivos `programa.c` y `foo.c`, existe una función `main()`. Por qué da error al intentar ejecutar la siguiente línea:

```
gcc programa.c foo.c
```

Ejercicio 5 - Hamming (2 puntos)

Explique brevemente qué se entiende por un algoritmo de corrección de errores y cual es su objetivo.